Digital Analog Design: A Highly-Efficient Method to Design Analog Circuits*

Mark Horowitz and Byong Lim
Department of Electrical Engineering,
Stanford University
Stanford, CA, 940305

Abstract: The past 30 years have seen an enormous growth in the power and sophistication of digital design tools, while progress in analog tools has been much more modest. Digital tools use many abstractions to allow them to validate implementations match the functional models, and the composition of cells matches the composition of the functional models. While there are many reasons why this is more difficult for analog circuits, it can be done. This papers shows how to leverage the smooth nature of the result surface of analog designs to accelerate the analog design and validation process.

Introduction

Unlike digital systems where scaling has been a win for performance, power, and cost, for analog designs, reducing supply voltages reduces maximum signal energy, making high SNR systems harder, and scaled MOS transistors have worse matching property and noise. As a result, it is not possible to build a pure analog system that meets today's system specifications. To achieve the required level of performance, current analog systems use many digital correction loops to improve the performance of the analog system [1]. These can be as simple as an offset correction loop for a comparator, to a complex digital correction of nonlinear amplifiers in an A/D converter.

This tight coupling of complex analog systems to complex digital solutions creates great challenges for chip validation groups because of the differing goals of analog and digital validation. Analog designers in general do not trust models; they validate what really matters – the circuits that are actually implemented. Since the output(s) are an analog (a.k.a. smooth) function of the inputs, analog designers focus on trying to ensure that the design meets its parametric goals, in terms of gain, bandwidth, noise, power supply rejection ratio, etc. They use accurate transistor-level models, and circuit simulators such as SPICE that solve large coupled nonlinear differential equations. While these simulations take significant time per run, this is not an insurmountable issue, since there are not many runs needed for validation.

On the other hand, the digital validation engineer need to ensure this complex digital system really works. Since the system is discrete (a.k.a. not smooth), they must run millions/billions of input vectors into it, and check to ensure it functions properly in all these cases. Given the large number of input vectors that need to be simulated, and the large number of gates in the final implementation, digital designers left simulating transistors and gates many years ago. Instead they design and validate at a higher level, using SystemVerilog or VHDL to create functional models of the design, and use formal checking tools like Formality [2] to ensure that the implementation matches the model.

Thus coupling an analog block into a complex digital validation is difficult. Transistor-level simulation of the analog part is generally out of the question, since it is too slow. To handle this issue, real number models of analog circuits in Verilog have been popularized for the verification of mixed-signal designs [3-5]. These models use discrete-time, real number values to represent the analog signals at the pins of analog blocks. While this approach is a great step forward, two principle issues remain: creating and validating these functional models.

To address these issues, we have been working on formalizing the approach to analog design for mixed signal systems [6] which we call "Digital Analog Design." It tries to take some of the productivity aids used in digital design/validation and adapt them to analog design. The overall approach is described in the next section.

Digital Analog Design

Given how critical and time consuming top-level validation is for digital designers, we assume that they will drive the full system validation effort. This means that we need to create an analog design approach that will both satisfy the requirements of analog design and fit into the digital validation flow. To us, this means that the system will need to use high-speed functional models for validation of the analog blocks at the top level, since nothing else will be fast enough. The key question is how to do this in a way that provides the same guarantees as we get with using high-speed functional models of the digital blocks.

Our methodology starts with partitioning a mixed-signal system into many smaller blocks by designers, each becoming an analog cell in the final design. Designers initially create simple functional models for these blocks to explore the design space. In our design approach, these models are written in SystemVerilog such that it can be used for validation. After roughly determining the requirements for each of the cells, designers then work at

* Much of this paper is from reference [13]

the transistor level to create the design that achieve these results for each of these analog cells. Since there is not a universal set of electrical rules checks (ERC) for analog cells, designers also create a set of ERC checks for each cell to check the environments where the cell is used. During this process, all the circuit-level issues are verified at block level in SPICE simulations, ensuring those issues will not break the functional model.

After the transistor level designs are complete, we essentially have two descriptions of each analog cell. To ensure the validation engineer can use the functional model with confidence, we need a way to formally compare it to the circuit implementation. This requires us to create a formal definition of an analog function. This abstraction is critical, since it not only helps us formally compare the functional model to the circuit implementation, but it will also guide us in the creation of analog functional models. For digital circuits, for instance, formal checkers use both Boolean value and synchronous time abstractions for the comparison.

Surprisingly, a piecewise linear model which captures its deviations from linear behavior (to capture weakly nonlinear behavior of the system) is the correct analog abstraction for the verification of analog systems. Remember, what makes a circuit analog is that there is a smooth relationship between inputs and outputs, and that in nearly all cases the designer wants this smooth relationship to be approximately linear. The importance of this linear (piecewise linear) behavior is brought home by the fact that the optimization objectives of analog circuits are either the properties of a linear system (e.g., gain and bandwidth) or the quantities that describe the deviation of the circuit from the linear model that we want to minimize (e.g., input offset of an amplifier, INL/DNL of a data converter, and signal distortion of a RF mixer during the frequency conversion).

An obvious objection to this claim is that no system is completely linear and many systems, such as variable gain amplifiers or PLLs, do not seem linear at all. While many of these systems are "nonlinear" when viewed in the current, voltage, and time domains, these systems can be transformed into linear systems by variable domain transformation [9]. For example, Figure 1 shows a block diagram of a typical high-speed link receiver and notes the variable domain in which each block could be mapped to a linear system model.[2] A PLL is highly nonlinear from a voltage-time perspective, but can be modeled as a linear system in phase domain of the clock input and output.

[2] The output of a slicer is a logic signal, which seems to be tricky to find its linear system model. This kind of an A/D conversion circuit needs a special transformation of the output to the input, resulting in a linear model which tracks the input threshold of the original circuit. The detailed description is found in [10].

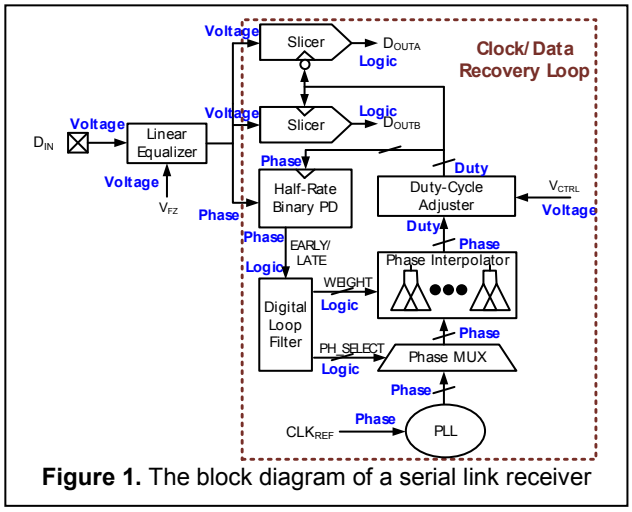

Figure 1. The block diagram of a serial link receiver

A variable gain amplifier is representative of another class of nonlinear circuits, where the "linear" output response is modified by the value of a control input; in this case it is the amplifier's gain. However, these kinds of circuits are easily handled by a slight extension of our linear model, since the system remains linear from input to output when the parameter controlled by the "control" inputs (e.g., gain) is fixed or varies slowly enough that they do not interfere with the main signal path. In these cases we characterize the linear system as a function of the control input. This characterization accurately models everything except the coupled dynamics between the input and the control. In fact, since dynamically coupled input/control interactions are hard to analyze and often lead to unwanted effects (intermodulation output terms), systems are always built where the control loop bandwidth is much slower than the signal path to avoid these effects. Thus, the relationship from the control input to the resulting parameter can also be generally modeled as a linear system as well. On reflection, the fact that a (piecewise) linear model is a good abstraction for analog circuits makes sense, since we tend to build systems that we know how to analyze, and these are systems whose behavior is linear in some domain (e.g., voltage for amplifiers, phase for PLLs, and frequency for mixers).

Note that this linear analog abstraction is primarily used for model comparison and help in understanding the circuit behavior for writing functional models. This does not mean that a Verilog model must be written as a linear system model in a possibly transformed variable domain. For example, it is more natural to write a phase detector model with a couple of flops and a few gates with the analog signals represented by the timing of the output pulses than to write it in phase domain. Similarly, while a class-D amplifier can be viewed as a linear system in duty-cycle domain, it can be straightforward to model it as a circuit whose switches are controlled by a periodic, pulse-width-modulated input signal. The abstraction's function is that it still defines what needs to be measured to ensure that the two descriptions match.

Our functional models are event-driven to enable efficient simulation in an event-driven logic simulator like SystemVerilog. It implies that each time the input changes, the model evaluates and updates the output waveform. The linear abstraction of analog circuits provides some clues on how to create good analog functional models that fit nicely into a digital validation context. We write a functional defines what needs to be measured to ensure that the two descriptions match.

Functional Model Validation

Our abstraction of analog circuits provides a formal way to compare two analog representations (e.g., a transistor schematic and an analog functional model): extract the function from each representation, and check to see if the functions match. If they do, the representations match, otherwise there is a mismatch. Since linear systems are characterized by a transfer function, (and possible distortion terms if deviation from linearity is critical in this application), the transfer matrix from the system's inputs to its outputs are extracted from both representations. For piecewise linear models, or units with controllable parameters, this matrix is extracted for the different operating points and corresponding matrixes are compared. If the extracted matrices do not match, then we can use the differences in the matrices to fix this mismatch by either changing the model or the implementation.

Surprisingly, automatically generating the test stimuli needed to extract this characterization of the analog blocks is not difficult. In digital systems, the response surface is completely discontinuous, which means one needs to explore all possible input values to explore its function. Therefore, the hardest part of such validation is how to generate test vectors to cover the complete result space quickly. For analog systems, the response surface is smooth, and can be explored with a few samples. With the linearity assumption, the surface is almost hyperplane, so theoretically we can characterize the output response of a circuit using only N+1 different input stimulus where N is the number of analog inputs. Although there are digital inputs as well in digitally-assisted analog circuits, many of those inputs have analog intent (e.g., quantized inputs), such that the number of test vectors still grows linearly with the number of inputs. True digital inputs, like power-down or calibrate, change the underlying circuit, so the transfer function of each circuit configured by true digital inputs must be compared between the two implementations [11]. In practice, we measure more than the minimal number of inputs vectors for each circuit for two reasons: to ensure that the extracted models are good estimates of the desired parameters in the presence of noise in the measured analog quantities and to generate more accurate models (e.g., weakly nonlinear models).

Figure 2 describes an overview of model checking procedure given that the port intents are labeled and the test setup for running simulations is prepared. Test vector

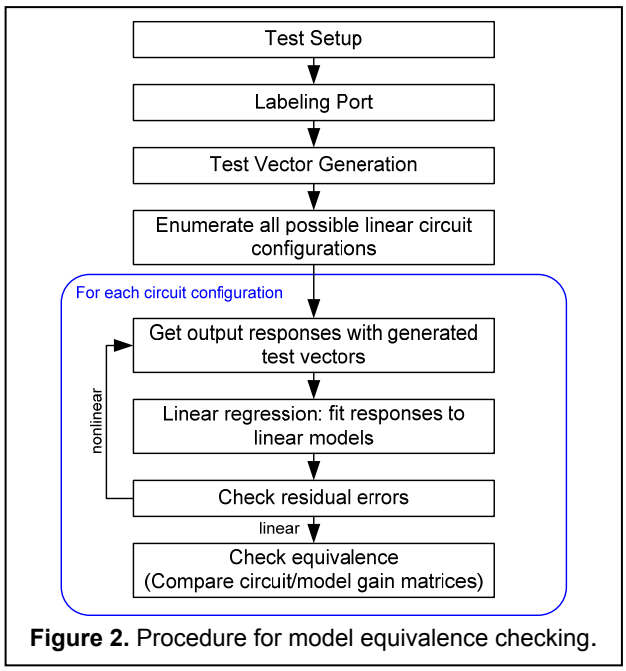

Figure 2. Procedure for model equivalence checking.

generation starts with enumerating all possible linear circuit configurations created by true digital inputs. The test vectors of N analog/quantized inputs are generated by Design of Experiment and used for all created circuit configurations. For robustness to simulation noise and model fitting errors, the number of test vectors is more than the minimum. The generated vectors are exercised on the circuit netlist and SystemVerilog model, and the transfer matrices of both models are then extracted and compared by performing linear regression on the response samples of analog outputs to analog/quantized inputs.

The challenge in analog model validation is that no pair of corresponding values in transfer matrices for two extracted systems has the same values. Even if the same system is validated, the extracted values will be different with different test vectors because of the inherent nonlinearity of circuits and numerical errors in simulation. This issue is handled by creating an allowable tolerance in the extracted parameters. Small mismatches simply indicate that the functional model does not represent the current circuit performance. Since these models are generally parameterized, the common approach is to extract the current values from the implementation, and use those in the functional models. In fact it is even possible to provide distributions of these parameters for the analog blocks, and perform Monte-Carlo like simulations at the functional level [11].

This validation method of a functional model is implemented as a tool, and various analog cells are tested with the tool [10, 11]. For example, Figure shows a block diagram of a single-slope ADC which adopts our “Digital Analog Design” flow. The functional models of all the blocks in a PLL (e.g., a phase frequency detector, a charge-pump circuit with a loop filter, a ring oscillator, and a

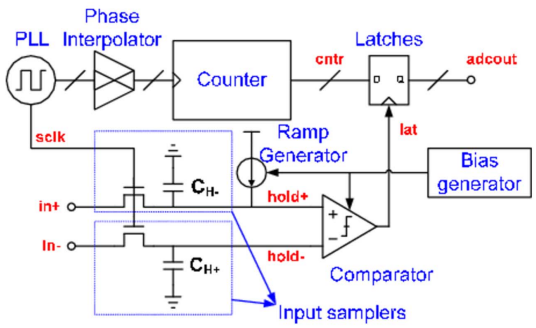

Figure 3. Block diagram of a single-slope ADC as a functional model validation example.

frequency divider) and other sub-blocks in the figure are checked against their circuit implementations to detect pin connection errors, and their parameters are updated from the checking results. Of course digital blocks (i.e., a counter and latches) are validated using a digital validation tool. The detailed results are found in [11], and other examples (i.e., sub-blocks of a serial link receiver) are found in [10].

Conclusion

Validation of complex mixed signal chips is forcing a change in analog design methodology. To be able to perform full system validation, high-performance, functional models of the analog blocks are required, and these models must be validated against the transistor level implementations of these blocks, if the system level validation is going to be meaningful. Digital analog design, which creates an event-driven, SystemVerilog functional model of analog circuits, and then validates these models against the transistor level implementation of this circuit, is an approach to address this problem. It uses a piecewise (mostly) linear abstraction of analog circuits to both guide the representation of analog signals in an event driven functional model, and to allow one to formally compare two analog descriptions. While this linear model initially seems very limiting, by transforming the circuit into the right domain, and leveraging the ability to handle piecewise linear, controlled systems, and distortion, we have not found a system that cannot be accurately modeled this way. In hindsight, the use of a linear abstraction makes sense, since it is the Lingua Franca of analog design. We are currently working on creating functional model templates for a number of analog blocks, to make the method even easier to apply.

Acknowledgements

This work was supported by Stanford SystemX, and the DARPA CRAFT program.

References

[1] B. Murmann, "Digitally Assisted Analog Circuits," *Micro, IEEE,* vol. 26, pp. 38-47, 2006.

[2] Synopsys, "Formality User Guide," ed, 2013.

[3] J. B. David, "Radio receiver mixer model for event-driven simulators to support functional verification of RF-SOC wireless links," in *Behavioral Modeling and Simulation Conference (BMAS), 2010 IEEE International*, 2010, pp. 42-47.

[4] J. E. Chen, "A modeling methodology for verifying functionality of a wireless chip," in *Behavioral Modeling and Simulation Workshop, 2009. BMAS 2009. IEEE*, 2009, pp. 96-101.

[5] Y. Wang, C. Van-Meersbergen, H. W. Groh, and S. Heinen, "Event driven analog modeling for the verification of PLL frequency synthesizers," in *Behavioral Modeling and Simulation Workshop, 2009. BMAS 2009. IEEE*, 2009, pp. 25-30.

[6] M. Horowitz, M. Jeeradit, F. Lau, S. Liao, B. C. Lim, and J. Mao, "Fortifying Analog Models with Equivalence Checking and Coverage Analysis," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, 2010, pp. 425-430.

[7] M. van Ierssel, H. Yamaguchi, A. Sheikholeslami, H. Tamura, and W. W. Walker, "Event-Driven Modeling of CDR Jitter Induced by Power-Supply Noise, Finite Decision-Circuit Bandwidth, and Channel ISI," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* vol. 55, pp. 1306-1315, 2008.

[8] M. H. Perrott, "Fast and Accurate Behavioral Simulation of Fractional-N Frequency Synthesizers and Other PLL/DLL Circuits," in *Design Automation Conference, 2002. Proceedings. 39th*, 2002, pp. 498-503.

[9] J. Kim, K. D. Jones, and M. A. Horowitz, "Variable Domain Transformation for Linear PAC Analysis of Mixed-Signal Systems," in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, 2007, pp. 887-894.

[10] B. C. Lim, J. Kim, and M. A. Horowitz, "An Efficient Test Vector Generation for Checking Analog/Mixed-Signal Functional Models," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, 2010, pp. 767-772.

[11] B. C. Lim, "Model Validation of Mixed-Signal Systems," Ph.D., Department of Electrical Engineering, Stanford University, 2012.

[12] J.-E. Jang, M.-J. Park, D. Lee, and J. Kim, "True Event-Driven Simulation of Analog/Mixed-Signal Behaviors in SystemVerilog: A Decision-Feedback Equalizing (DFE) Receiver Example," in *Custom Integrated Circuits Conference (CICC), 2012 IEEE*, 2012, pp. 1-4.

[13] B. C. Lim, J. E. Jang, J. Mao, J. Kim and M. Horowitz, "Digital Analog Design: Enabling Mixed-Signal System Validation," in IEEE Design & Test, vol. 32, no. 1, pp. 44-52, Feb. 2015.